Banco de dados II

Profº Jânio Eduardo

janio.vasconcellos@gmail.com

Roteiro

- Objetivo geral
- Introdução a banco de dados
  - Revisão de modelo conceitual;
  - Revisão de modelo lógico e físico;
  - Criação do ambiente com XAMPP;
  - Apresentação do PHPMyAdmin;
  - Criação de tabelas com PHPMyAdmin;
  - Inserção de dados com PHPMyAdmin;
- Resumo da aula

Estrutura de um select

Select * from _tabela

Select

- O comando SELECT é utilizado para extrair os dados das tabelas de um banco de dados. Ele pode extrair dados de uma ou mais tabelas ao mesmo tempo, executando desde simples consultas até comandos mais complexos, fazendo buscas, junções, filtros comparativos, ordenações e diversos outros itens. Abaixo, mostraremos algumas das principais cláusulas para o uso do comando SELECT

Comando select

- SELECT * FROM nome_da_tabela;

Cláusula WHERE

- Agora que já sabemos extrair os dados de nossas tabelas de forma simples, podemos começar a adicionar filtros para nossas consultas. A cláusula **WHERE** é a responsável por filtrar resultados, utilizando parâmetros comparativos como **igual, diferente, maior, menor e entre outros**. Além disso, podemos usar diversos filtros em nossas consultas, inclusive adicionar parênteses para priorizar consultas dentro de escopos. A cláusula WHERE funciona da seguinte forma:

- SELECT coluna, coluna FROM nome_da_tabela WHERE condicao_da_consulta;

Principais comparativos

- Além do comparador de igualdade “=” (exemplificado acima), podemos utilizar diversos outros tipos de comparadores para melhorar a nossa consulta.
- **Operador comparativo “<>” (diferente)**
  - Para buscar itens que sejam diferentes de algo, podemos utilizar o comparador de diferença “<>”:

Operador comparativo “>” (maior que)

- Para buscar itens que sejam maiores que algo, podemos utilizar o comparador “>”:
- Essa busca retornará todos os itens que possuem a data maior que 16/09/2016.
- Uma possível variação do operador ‘maior que’ é o operador ‘maior ou igual a’, representado pelos caracteres “>=”. Se usássemos o operador ‘maior ou igual a’ no exemplo acima, a consulta iria buscar os posts com data maior que 16/09/2016 e os posts que possuem exatamente essa data.

Operador comparativo “<” (menor que)

- Ao contrário do operador apresentado anteriormente, o operador ‘menor que’ irá buscar itens que sejam menores que algo. O operador é representado pelos caracteres “<”. Veja um exemplo de comando com esse operador abaixo:
- Essa busca retornará todos os itens que possuem a data menor que 16/09/2016. Da mesma forma que o operador apresentado anteriormente. O operador ‘menor que’ também possui a variação ‘menor ou igual a’.

Operador comparativo “LIKE” (busca parcial de texto)

- Para buscar itens que contenham trechos de um texto, podemos utilizar o comparativo LIKE. O comparativo LIKE vem acompanhado de uma máscara de filtro. Essa máscara pode alterar de um SGBD para o outro, nesse exemplo utilizaremos a máscara “%”.
- Se por exemplo quisermos buscar uma postagem em que o título inicia com a palavra “Guia”, podemos buscar da seguinte forma:

Juntando comparativos

Operador lógico “AND”

- O operador “AND” verifica as condições como um todo e retorna os resultados que atendem a todas as condições. Para buscar postagens que possuem o status 1 “**E**” a data maior que 21/09/2016, utilizaremos o comando:

Operador lógico “OR”

- O operador “OR” verifica separadamente as condições e retorna os resultados que atendem pelo menos uma das condições. Para buscar postagens que possuem a data maior que 17/10/2016 **“OU”** menor que 09/09/2016, utilizaremos o comando:

Uso de parênteses

- Os parênteses são utilizados para dar prioridade a uma condição dentro de uma consulta. Se necessitarmos buscar postagens que foram feitas após o dia 17/10/2016, independentemente de seu status e, além disso, buscarmos notícias mais antigas somente com o status ativo, faremos da seguinte forma:

Cláusula ORDER BY

- Hora de ordenar os nossos resultados com a cláusula ORDER BY. Inicialmente, vamos ordenar de modo ascendente (ASC) e depois de forma descendente (DESC). A cláusula ORDER BY deve ser inserida após as cláusulas anteriores. Não ficou claro? Então dá uma olhada em nossos próximos exemplos!

Modo ascendente (ASC)

- O modo ascendente (ASC) é o padrão do ORDER BY. Portanto, nem é necessário estar explicito no comando. Contudo, por razões didáticas, iremos mostrar dois comandos que são diferentes, mas que retornam os mesmos resultados devido essa particularidade do modo ascendente (ASC).

Modo descendente (DESC)

- Em blogs, a última postagem é sempre a que fica em primeiro lugar na lista. Para essa funcionalidade, utilizaremos a ordenação descendente através da data como referência:

Junção de tabelas

- Parabéns, você chegou até aqui, aposto que você já estourou a sua quota de conhecimentos pelo dia, mas para ficar completo, ainda há mais conteúdo pela frente. Então faz um esforço final que já está nos ‘finalmente’. Bom, agora que você já sabe buscar, filtrar e ordenar nossos resultados, chegou o momento de unirmos diferentes tabelas em uma só consulta. Um exemplo de aplicação desse recurso seria trazer postagens do blog juntamente com as suas imagens relacionadas.
- Para isso, no comando SELECT, precisamos utilizar o prefixo das tabelas para especificar quais campos desejamos trazer.

Junção de tabelas

- Para exemplificar, vamos fazer uma consulta simples utilizando duas tabelas. Utilizaremos nossa tabela *blog_posts* e *blog_post_image* para descobrir quais imagens estão relacionadas com cada postagem. Nessa consulta, retornaremos o id e título da postagem, além do id das imagens relacionadas. Veja o comando a seguir:

```
SELECT nome_da_tabela.campo_da_tabela FROM nome_da_tabela,
nome_da_tabela_2 WHERE condicao;
```

Aliás e Tabelas

- É importante perceber que na cláusula FROM colocamos todas as tabelas que são referenciadas. Cada vez que precisamos utilizar uma coluna, precisaremos acrescentar como prefixo o nome da tabela. Para consultas maiores, isso pode ficar muito repetitivo. Pensando nisso, podemos usar um “alias”. O aliás (ou pseudônimo) serve para criar uma chave única que referencia a tabela, facilitando o entendimento e a manutenção das consultas. A consulta acima, utilizando pseudônimos (alias), ficaria assim:

Obrigado!!!

Profº Jânio Eduardo

janio.vasconcellos@gmail.com
(61) 98451-9188